

AD-A284 575



CDRL: B013
22 February 1994

UNISYS

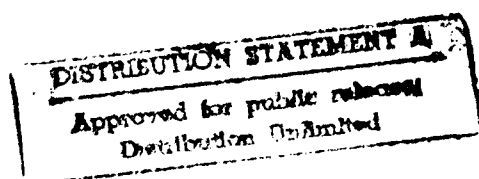
**Command Center Component
Supported Component Report**
Central Archive for Reusable Defense Software
(CARDS)

Informal Technical Report



Central Archive for Reusable Defense Software

STARS-VC-B013/001/00
22 February 1994



94-29951



DTIC QUALITY INSPECTED 5

CDRL: B013
22 February 1994

**INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)**

*Command Center Supported Components Report
Central Archive for Reusable Defense Software
(CARDS)*

**STARS-VC-B013/001/00
22 February 1994**

Data Type: Informal Technical Data

**CONTRACT NO. F19628-93-C-0130
Line Item 0002AB**

Prepared for:

**Electronic Systems Center
Air Force Material Command, USAF
Hanscom AFB, MA 01731-2116**

Prepared By:

**Unisys Government Systems Group
Information Systems
12010 Sunrise Valley Drive
Reston VA 22091**

**Distribution Statement "A"
per DoD Directive 5230.24
Approved for public release, distribution is unlimited**

CDRL: B013
22 February 1994

INFORMAL TECHNICAL REPORT
For The
SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS
(STARS)

Command Center Supported Components Report
Central Archive for Reusable Defense Software
(CARDS)

STARS-VC-B013/001/00

Data Type: Informal Technical Data
22 February 1994

CONTRACT NO. F19628-93-C-0130
Line Item 0002AB

Prepared for:

Electronic Systems Center
Air Force Material Command, USAF
Hanscom AFB, MA 01731-2116

Prepared By:

Unisys Government Systems Group
Information Systems
12010 Sunrise Valley Drive
Reston VA 22091

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

CDRL: B013
22 February 1994

Data Reference: STARS-VC-B013/001/00
INFORMAL TECHNICAL REPORT
Command Center Supported Components Report

Distribution Statement "A"
per DoD Directive 5230.24
Approved for public release, distribution is unlimited

Copyright 1994, Unisys Corporation, Reston, Virginia

Copyright is assigned to the U. S. Government, upon delivery thereto, in accordance with the DFARS Special Works Clause

This document, developed under the Software Technology for Adaptable, Reliable Systems (STARS) program, is approved for release under Distribution "A" of the Scientific and Technical Information Program Classification Scheme (DoD Directive 5230.24) unless otherwise indicated. Sponsored by the U. S. Advanced Research Projects Agency (ARPA) under contract F19628-93-C-0130 the STARS program is supported by the military services with the U. S. Air Force as the executive contracting agent. The information identified herein is subject to change. For further information, contact the authors at the following mailer address: delivery@stars.reston.paramax.com.

Permission to use, copy, modify, and comment on this document for purposes stated under Distribution "A" and without fee is hereby granted, providing that this notice appears in each whole or partial copy. This document retains Contractor indemnification to the Government regarding copyrights pursuant to the above referenced STARS contract. The Government disclaims all responsibility against liability, including costs and expenses for violation of property rights, or copyrights arising out of the creation or use of this document.

The contents of this document constitutes technical information developed for internal Government use. The Government does not guarantee the accuracy of the contents and does not sponsor the release to third parties whether engaged in performance of a Government contract or subcontract or otherwise. The Government further disallows any liability for damages incurred as the result of the dissemination of this information.

In addition, the Government (prime contractor or its subcontractor) disclaim all warranties with regard to this document, including all implied warranties of merchantability and fitness, and in no event shall the Government (prime contractor or its subcontractor) be liable for any special, indirect, or consequential damages or any damages whatsoever resulting from the loss of use, data, or profits, whether in action of the contract, negligence, or other tortious action, arising in connection with the use or performance of his document.

CDRL: B013
22 February 1994

Data Reference: STARS-VC-B013/001/00
INFORMAL TECHNICAL DOCUMENT
Command Center Supported Components Report
Central Archive for Reusable Defense Software (CARDS)

Principal author of document:

James Estep

Kerri Haines

Approvals:

System Architect *Kurt Wallnau*

Program Manager *Lorraine Martin*

(signatures on File)

Data Reference: STARS-VC-B013/001/00
INFORMAL TECHNICAL REPORT
Command Center Supported Components Report

ABSTRACT

A software reuse library is only useful if the assets within the library can be quickly and easily located, evaluated, and the characteristics of the assets be easily understood. This Command Center Supported Components Report provides a set of guidelines, initially developed for Command Center Libraries, to assist domain engineers and/or the qualification team in charge of the library in providing information on domain-specific qualified library assets. These guidelines consist of one page templates describing a component from a high level viewpoint, i.e., "glossies", and a multi-page template describing a component in greater technical detail, i.e., "technical brief".

Two examples of a glossy and technical brief are provided in the appendix.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 22 February 1994	3. REPORT TYPE AND DATES COVERED Informal Technical	
4. TITLE AND SUBTITLE Command Center Supported Components Report CARDS			5. FUNDING NUMBERS F19628-93-C-0130	
6. AUTHOR(S) James Estep Kerri Haines				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Unisys Corporation 12010 Sunrise Valley Drive Reston, VA 22091			8. PERFORMING ORGANIZATION REPORT NUMBER STARS-VC-B013/001/00	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Department of the Air Force Headquarters Electronic Systems Center Hanscom AFB, MA 01731-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER B013	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Distribution "A"			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) See Abstract Page				
14. SUBJECT TERMS			15. NUMBER OF PAGES 28	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	

Table of Contents

1	Introduction	1
1.1	Overview	1
1.2	Audience	1
1.3	Concept	2
2	Background	2
2.1	Component Qualification and the Model-Based Approach to Reuse	2
2.2	Component Qualification and Preferred Products	3
2.3	Identifying Domain-Specific Library Components	4
2.4	Distribution and Modification Limitations/Restrictions	5
2.5	GOTS, COTS, and Public Domain relationships	6
3	Developing a Glossy	7
3.1	Role within the Domain	7
3.2	Salient Features	8
3.3	Technical Specifications	8
3.4	Support	8
3.5	Licensing and Availability	8
4	Developing a Technical Brief	10
4.1	Introduction/Overview	10
4.1.1	Application Area	10
4.1.2	Limitations	10
4.2	Detailed Functional Description	10
4.3	Specific Functions within the Domain(s)	11
4.4	Description of the Salient Features of the Component	12
4.5	Technical Specifications, Support, and Maintenance information	12
4.6	Licensing and Availability	12
5	Summary	13
	Appendix A	A-1
	Appendix B	B-1
	References	R-1

List of Figures

Figure 3-1. Glossy template	9
Figure 5-1. Cover for Technical Brief.....	14

Command Center Supported Components Report

1 Introduction

1.1 Overview

This report describes informational and technical documents intended to describe qualified components that are assets within a domain-specific reuse library[7]; These documents are intended to be used by library users (engineering personnel) who need to access components stored within the library and by potential component providers. Two different types of documents are described: "*glossies*" and "*technical briefs*". The purpose of this document is to describe what information these reports contain, their intended use, and the manner in which they are created. Samples of these reports are provided for illustrative purposes.

Glossies are single page documents describing a component within a domain-specific library from a top-level perspective. The technical brief is a multi-page document describing the component in greater detail, to include its differences relative to other library components, and the results of the qualification process through which the component qualified for inclusion in the library. Both serve as an application guide for the customer of the library, providing information on "form, fit, and function" measures of compatibility between software components and the system or domain architecture. Together they provide the library user a mechanism for identifying "architecture-compliant" components for building a system.

Examples of specific glossies and technical briefs are provided for two qualified components stored within the Central Archive for Reusable Defense Software (CARDS) Command Center Library (CCL). The components are OILSTOCK, a high resolution interactive graphics system, and the Software Engineering Institute's MTV, a message translator and validator. Although these examples provided are specific to the Command Center Domain, the details for creation of glossies and technical briefs are equally applicable to any domain.

1.2 Audience

This *Command Center Supported Components Report* is intended to be used by the domain-engineers and/or qualification team (responsible for qualifying components into a domain specific reuse library) for preparing glossies and technical briefs for library components. The qualification team in collaboration with the component vendor will create glossies and technical briefs (or modify existing information provided by component creators) containing domain-specific qualification information. The qualification team will use the Product Evaluation Report (PER), a by-product of the component qualification process, for particular products to aid in completing the glossies and technical briefs [3]. The vendor may also use the PER to enhance future versions of their product as well as aid in preparing the information they provide for the glossy and technical brief.

The glossies and technical briefs will be used by engineers, engineering managers, and component vendors seeking to access components stored within a domain-specific model-based reuse library [4].

1.3 Concept

The concept guiding the development of both templates focuses on providing the library user with a clearer understanding of the component—system—architecture—domain relationship. Providing the library user with a medium for framing the relationship of a component to a particular domain is very important for effectively and efficiently building systems. A component fulfills a role within a system which fulfills a role within an architecture and ultimately within a domain. These roles can only be fulfilled if the necessary constraints and requirements of the domain are met. Clearly communicating these constraints and requirements via supplementing library services is a main objective of the glossy and technical brief templates.

A closely related objective of the glossy and technical brief templates is to provide a vehicle for components to be advertised or marketed to library users. The marketing environment introduced by the glossies and technical briefs provides a motivation for future growth of the library because component providers will need to keep their products up to date and competitive with the other components to ensure they will be chosen for system development.

2 Background

2.1 Component Qualification and the Model-Based Approach to Reuse

There are generally two approaches to implementing a reuse library [9]. One is a *component-based* approach and the other is a *model-based* approach. Component-based libraries are organized around collections of reusable components (e.g., software, documentation, and architectures). While component-based libraries can support reuse of a broad-spectrum of component types, the underlying operational concept is that of search and retrieval of individual, usually unrelated, components. When components are inserted into these libraries, they are typically classified in broad, generalized categories; information describing their specific role in a particular problem-domain architecture is not formally encoded. For example, there is no schema for describing the effectiveness or efficiency of a component for any given domain (e.g., command center). As a result, information about domain-specific usage context is no longer attached to the component and therefore lost to the reuser. Advantages of a component-based approach include strong search mechanisms based on well-known classification techniques, the ability to reuse across a broad range of application areas simultaneously, and scalability of the mechanism to very large component populations.

Model-based libraries use domain models as a foundation for library organization and a framework for supporting applications which exploit these models to automate various library services. The library model encompasses information such as domain knowledge, generic architecture specifications, requirements, implementation restrictions, as well as software arti-

facts (including off-the-shelf products). The inclusion of this additional information within the library model supports a "components in — subsystem out" paradigm that facilitates the concept of mega-programming desired by the Department of Defense [8]. Advantages of a model-based approach include the ability to provide a domain-context for components and support for reuse-oriented engineering activities (e.g., system composition, component qualification). Disadvantages include weak associative search and retrieval, a lack of cross-domain component search and retrieval, and relatively small component population resulting from a narrow domain focus.

While the CARDS Command Center Library (CCL) adheres to a model-based paradigm in support of domain-specific reuse, the nature (as illustrated by the advantages and disadvantages) of the approaches suggest that they are not diametrical, but rather complementary [9].

In summary, CARDS distinguishes the concept of a Component-based library from a Model-based library [7]. The CARDS model-based approach, uses a library model which encodes and describes the relationship of the components in the library against the requirements and constraints imposed by a specific application domain (e.g., command center). These constraints form an evaluation criteria for components evaluated for inclusion in the library. The glossy and technical brief provide user-friendly marketing information on the results of the component qualification process.

2.2 Component Qualification and Preferred Products.

The content described in the glossy and the technical brief is supported and facilitated by the information gathered during the Component Qualification Process[3] [6]. Component qualification is the process of acquiring and evaluating components for a domain-specific library. The key role of component qualification is to measure the "form, fit, and function" of a component against the constraints inherent within the architecture for that domain. Components are measured against domain and common metrics in the evaluation process. Domain requirements are established in a high-level form, for example, the Portable, Resuable, Integrated Software Modules (PRISM) program's command center prototype. These domain requirements are then mapped to features of the component family to establish a set of component constraints. These component constraints, along with architectural and implementation constraints, make up the domain metrics. Common metrics are criteria used to evaluate components regardless of domain. Common metrics are defined based on categories such as reliability, maintainability, portability, and security. Results of the evaluation process will be available to users and may indicate the need for wrapper software to fully implement the requirements of the domain.

Commercial software certified and qualified against criteria specific to a domain would be listed as products or components which have been distinguished as *preferred products* and would be considered better suited for systems development (relative to other commercial products which have not been pre-evaluated for systems development) in that domain.

The glossy and technical brief templates discussed in this report play an important role in supporting this concept by providing a vehicle to "market" CCL architecture-compliant components (or any libraries components for that matter). Other reuse libraries, such as Defense Information Systems Agency/CIM's Defense Software Repository System (DISA/CIM's DSRS) [1] and the Software Technology for Adaptable, Reliable Software Asset Source for Software Engineering Technology (STARS ASSET) [5] reuse libraries have established certification processes which play an important role in this vision.

2.3 Identifying Domain-Specific Library Components

Once a contractor is chosen to develop a system, the losing bidders proposed system(s) (and thus proposed components) are most often "shelved". Unique and valuable ideas, designs, paradigms, or actual implementations thus may not be utilized because they are "embedded" in the losing system. A bidder is often chosen because the Government decided that the development proposal and/or approach for the proposed *system* best met desired constraints and goals at a system level. There is very little focus on whether a bidder intended to use newly developed components or reuse existing components in the bidding process.

The reality of this scenario is that several of the components that comprise the competing systems may serve basically the same functionality. This "reinventing of the wheel" can mean a significant amount of time and money being wasted. It could also be that components from other previously developed or proposed systems would be superior to comparable components in the winning bidders system, but since the proposals are at a system level there is no mechanism for utilizing or evaluating superior components. From the Government's perspective, under this scenario, it may not be getting the optimal systems.

The likelihood that a large percentage of the components in the desired system have already been produced in previous systems is a very important issue. The issue must be addressed by eliminating this unnecessary duplication and waste of valuable resources. If the contractors could have chosen from a library of components or systems, tested and categorized for building systems in a specific domain — the valuable time, money, and resources could have been saved.

The Government increasingly requests proposals for a system based on the domain architecture. Using the "Command and Control (C2) Store" - a conceptual model for developing command centers, as a basis, components are evaluated and tested to be compliant with the domain architecture. The contractors evaluate components for building the system by consulting a domain-specific library. Specific components can be evaluated relative to system requirements using library services, i.e., browser services. During this evaluation process it is important that components can be easily and efficiently identified and analyzed. The standard formats for the technical briefs and glossies used in conjunction with other library services (on-line or in hard-copy) provide a familiar and easy to follow resource for evaluating components and fulfilling this requirement.

In other words, a significant percentage of a proposed system will be comprised of existing components and subsystems identified via a domain-specific library. This identification process is very important. A domain-specific library may have many valuable resources, but if they can not be identified and extracted with the least amount of effort the maximum intended benefits can not be realized. The contractor must have a sound and effective interface or "communication" mechanism with the domain-specific library to effectively identify components for system development. Competing contractors can then be evaluated by how they assemble existing off-the-shelf qualified components and by their designs for the customized percentage of the system that could not be developed with existing components.

If a contractor has a component or a plan for the implementation of a component that outperforms existing components, either that component can be submitted for qualification or the plan for development of the component can be submitted as part of the proposal. To further expand, if a new version of a component has been developed from an existing qualified component or from an unqualified component, the new version would need to be (re)qualified based on the added or modified functionalities. This process lessens the danger of impeding the development of newer and better components and subsystems.

The results are a cost effective proposal and an evaluation process which saves money and time in terms of proposal preparation. The Government can be assured that the majority of the components labeled in the system are the "best" available. In addition, less money and time will have to be expended in acquiring the desired system. The Government can then focus on evaluating the customizations proposed by each contractor which will be significantly smaller than evaluating a complete system proposal.

The concept described above may leave one important issue unclear. How do we maintain the originality and creativity needed to incorporate innovation and growth into future applications in the domain? It must be understood that this scenario does not forbid the improvement of existing systems. Nor does it limit the variation of a particular component. Since the nature of the new component acquisition process moves the focus to the component level, vendors of a particular component will need to continually maintain and improve the component or it will not be used in future system development.

2.4 Distribution and Modification Limitations/Restrictions

Each of the levels of rights associated with components or subsystems requires the software or technical data to be marked in a certain way and may also have certain use, distribution, and modification limitations/restrictions. The Government has either: unlimited, resuricted, limited, or Government Purpose License Rights (GPLR), (depending on what was negotiated in the original contract with the contractor) rights over the custom software systems it contracts for development (namely, government-of-the-shelf (GOTS))[2]. This type of information is very important when identifying components for development. A domain-specific library providing

components for system development must ensure that the library user can accurately identify any and all restrictions associated with a particular library asset.

Commercial-of-the-shelf (COTS) software also has defined restrictions. Ownership remains with the commercial vendor and specific agreements concerning usage and licensing are established. The commercial vendor may need to implement modifications to the component(s) to meet domain criteria and keep the component up to par with qualification standards. The commercial vendor assists library maintainers in preparing glossies and technical briefs based on the component evaluation. If the vendor for whatever reason does not make required modifications, the component may not be eligible for use in system development. This introduces an element of competitive leverage to the process. Which in turn provides the stimulation for maintaining quality components.

2.5 GOTS, COTS, and Public Domain relationships

The preferred product provider, i.e., commercial vendor, has a very unique opportunity. In an acquisition process, which focuses on acquiring discrete software components instead of complete systems, a particular commercial component vendor is now presented with a marketing mechanism for a product (e.g., Command Center Library). In the spirit of a "Preferred Product Vision", a list is provided of components readily available for building software systems; allowing smoother selection and providing justification for use of COTS (or GOTS) software. At the same time the Government is reducing risk and costs of systems development.

Currently, public domain software is usually available "as is". It is often the case that there is no way to indicate the level of quality, and thus the ease of maintenance and reliability. This often means a lot of time and resources in debugging and identifying necessary modifications to meet the needs of the intended system. The important point with public domain software is that once it has been certified, qualified, and placed in a library, it can be extracted without unnecessary and time consuming work to bring it up to standard.

Another important point is that just because a component may be Public Domain does not necessarily give the user the right to modify the code. Public Domain software can carry restrictions and limitations on distribution, use, and modifications [2]. Public Domain software is not usually sold, so there are no distributors or retailers. They are usually found by word of mouth or on public bulletin boards.

Once a public domain application has been certified (e.g., ASSET certification process [6]) and then qualified through the component qualification process (and a glossy and technical brief are prepared), the system developer can proceed with more confidence in the component.

Whether the component is COTS, GOTS, or Public Domain, the library maintainers will perform an analysis of the components architecture compliance to determine the degree to which it meets the requirements and constraints of the intended mission area (e.g., command center). If the components meet the criteria (at least with allowable modification, i.e., wrappers), they become a qualified component of the library. There may also may circumstances in which components are

excepted from other programs performing domain specific work (e.g. PRISM). A glossy and technical brief would then be prepared for the component to be used as an aid in the application development and component acquisition processes.

3 Developing a Glossy

A library component glossy provides a standardized format for the library user to easily gain information about a particular component. The library services provide the ability for the user to quickly identify the components applicability to their system design. The information provided by the glossy enhances this ability. It provides a platform for advertising the components capabilities within the specific domain to the library users as they browse the library or by printing hardcopies at a workshop, conference, or related event. The glossy may be accessed on-line via the library.

The realization of these abilities require glossy support of the following four uses or purposes:

1. Serve as a vehicle for communicating the major characteristics of domain-compliant components.
2. Provide a high level summary of the component qualification results.
3. Provide summary of technical details.
4. Support evolution of the library (e.g., preferred products list).

The glossy has a simple format intended to convey the appropriate amount of information needed for advertising the component without going into technical specifics. More technical details can be provided in the technical brief. It is recommended that the technical brief be completed first, then appropriate information can be extracted to complete the glossy. In so far as content, the glossy can be viewed as a subset of the technical brief. The main difference being that the audience viewing the glossy will probably be more concerned with a quick and easy to follow evaluation. Once the glossy has been used to identify a particular component more detailed information can be extracted from the technical brief. The template for the physical arrangement of the glossy can be found in Figure 4-1. As can be seen in Figure 4-1, the glossy should cover the following areas:

1. Role within the Domain
2. Salient Features
3. Technical Specifications
4. Support
5. Licensing and Availability

The content of the glossy information should be equivalent to one page (maximum), front and back. Each section should clearly present information to the customer.

3.1 Role within the Domain

The information to convey in this section should answer "What is the function of this component within a specific domain?" Basically this section should be based on the executive summary from the Component Evaluation [3].

3.2 Salient Features

This section should clearly define "What distinguishes the component from other components filling the same role within the domain?"

3.3 Technical Specifications

Define technical requirements that must be met for this component. Answering this question would involve a brief description of hardware and software requirements.

3.4 Support

Outline who currently supports this component. What are some points of contact for the component?

3.5 Licensing and Availability

Give licensing information involved with utilizing the component. What costs will be involved? What are the options for quantity of license (e.g., single or multiple user)? The information should include: intellectual property (who holds copyright, patent or trade secret and when was it issued), level of rights for the Government funded components (unlimited, limited, restricted, or Government Purpose License Rights), and associated limitations and/or restrictions on distribution, and modifications.

The Command Center Library has the legal obligation to supply information on intellectual property (i.e., patents, copyrights, trade secrets, and associated restrictions/limitations on use, distribution, and modification). By including this information in the glossy (as well as the technical brief) the library user can see it when accessing components while browsing the library.

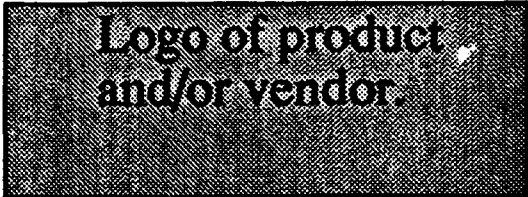
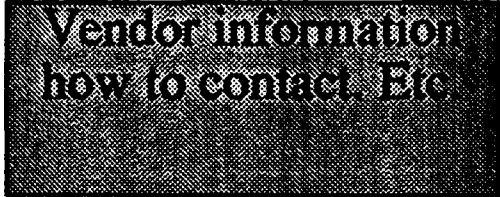


COMPONENT INFORMATION	
	
Role within the <fill-in> Domain	
 <p>Figure</p>	<p>Function within the specific domain? What is the CCL view? Is a demonstration available?</p>
Salient Features	
 <p>Figure</p>	<p>What distinguishes this component from other components that fill the same role within the domain?</p>
Technical Specifications	
<p>Technical requirements? Hardware/Software requirements?</p>	
Support	
<p>Point of contact? Who supports it?</p>	
Licensing and Availability	
<p>License and cost? Single or multiple users?</p>	

Figure 3-1. Glossy template

4 Developing a Technical Brief

The purpose of the technical brief is to provide a potential developer with a detailed description of a component in terms of a particular domain or set of applicable domains. The technical brief is an extension of the glossy. It covers the same information as the glossy only in much greater detail. The layout of the technical brief should be as follows:

1. Introduction/Overview
2. Detailed Functional Description
3. Specific Functions within the Domain
4. Description of the Salient Features of the Component
5. Support/Maintenance information

This layout is similar to the glossy layout because the technical brief expands upon the information in the glossy. An example of the technical brief can be found in the Appendix B. Figure 5-1 presents an example of the technical brief cover page.

4.1 Introduction/Overview

The overview section should answer the basic questions, "What is the component?" "Where did the component originate from?" The functionality should also be described, but briefly and clearly enough to be quickly understood and provide an introduction. In general, it should provide a functional overview. Appropriate application areas, limitations, and a discussion of the interface mechanisms should also be included in this overview.

4.1.1 Application Area

Describe the application areas within the domain for which the component was originally developed. List possible variations or alternate usage for the component.

4.1.2 Limitations

Describe known limitations of the component within the domain. Describe special circumstances in which the component may not perform as described in the functional description.

4.2 Detailed Functional Description

This section of the technical brief should provide a sufficiently in-depth discussion of the functionality of the component. The term "sufficiently" is used because the intent of the technical brief is to provide information about the component in terms of the particular domain. A fine granular description of the component can be found in the technical documentation for the component and references to these technical documents should be included. Information for this section should come from at least the following areas (as appropriate):

1. Interface functionality (e.g., signal or interrupt)
Interface communications (e.g., name, type, scope, use, range, etc.)
Effects of a call to the interface (i.e., describe as well as possible, refer to technical documentation for the component)
2. Memory management
3. Return values (e.g., type, description)
4. Side effects
5. Error handling
6. Dependencies (i.e., sequences that must be followed)
7. External subprogram calls and parameters
8. Areas of special attention (i.e., suggestions for future development)
9. Instantiation and Integration instructions
10. Testing/debugging instructions

4.3 Specific Functions within the Domain(s)

This section should contain information about the specific function that the component serves relative to the domain (or domains) in which it is applicable. It is very important that this section be clear because it is possible that a particular component may be capable of serving multiple purposes within a domain. If this is the case the discussion for each role should be kept separate and distinguishable. The description of the roles do not need separate technical briefs, but should be clearly labeled within the subsection of the technical brief.

A component is qualified against the features and functionality required of it for inclusion in the domain for which it is being evaluated. This set of required functionalities for the domain is used during the component qualification process to determine if the functionalities (or a subset thereof) qualifies it for one or more purposes within the domain. This implies that the component may have multiple sets of functionalities, fulfilling separate roles, for the domain in which it is being qualified. This "multi-set concept" obviously applies to different domains. For instance, a data base management system (DBMS) may be qualified for the command center domain because it has specific functionality defined by functionality set A. But, for the domain of say intelligence, that same DBMS may be qualified because it has specific functionality defined by functionality set B. Sets A and B may have an intersecting subset, but may not be proper subsets. It is these clearly defined sets that reflect the differing requirements for each of the respective domains (i.e., set A for command centers and set B for intelligence). It is the goal of this section to convey those requirement sets that are satisfied — in a clear and distinguishable manner.

The functionality of a component and its relationship to the domain are best described in the component class (e.g., SYBASE and INGRES are components that are "instances" of a DBMS in the context of the command center domain [3]). For any component in the command

center domain, the component must have met the criteria established for it in its component class (e.g., DBMS) before it could have been accepted. To “generically” look at the specific functions of a component in a domain, look to the definition of the component class in that domain. From the component class you can trace back to the requirements for a specific component feature defined in that component class (e.g., “the ability to specify queries utilizing ANSI SQL” is a feature of the component class of DBMS). To “specifically” see if a particular instance of a component class (e.g., SYBASE) satisfies or is qualified for some feature of that component class, trace the requirements of that instance within the component class (e.g., does SYBASE have the “ability to specify queries utilizing ANSI SQL”).

With this in mind, here are some questions to be answered when completing this section of the technical brief (note: These are not all inclusive.):

What function or feature set is required of the component for the particular domain?

Are there multiple function/feature sets for this component in this domain?

What are the other domains in which the component is qualified?

What features are optional for the component for the particular domain?

What requirements “lead” to those required component features?

4.4 Description of the Salient Features of the Component

This section defines the uniqueness of the component compared to other components that fulfill the same role within the domain. What features of a specific component are discriminators with which one might electively choose one component over another (e.g., if SYBASE and INGRES are equally qualified — why would SYBASE be chosen over INGRES?)? These are some questions that should be answered when completing this section:

what does the component do that nobody else’s does?

what does the component do better than anyone else’s component?

what should prospective customer’s remember about the product?

4.5 Technical Specifications, Support, and Maintenance information

Provide information detailing how to contact support personnel concerning questions related to the component. You should provide a separate section in the technical brief for each of these items. This information should include: names, telephone numbers, conventional mail addresses, and email addresses. In addition, if possible, include hours, a fax number, and alternate electronic service addresses (e.g., Compuserve). Other important information includes consulting and training information, licensing information, and ordering information.

4.6 Licensing and Availability

The information found in this section of the technical brief should cover basically the same information found in the corresponding section of the glossy.

The information should include: intellectual property (who holds copyright, patent or trade secret and when was it issued), level of rights for the Government funded components (unlimited, limited, restricted, or Government Purpose License Rights), and associated limitations and/or restrictions on distribution, and modifications.

The Command Center Library has the legal obligation to supply information on intellectual property (i.e., patents, copyrights, trade secrets, and associated restrictions/limitations on use, distribution, and modification). By including this information in the glossy (as well as the technical brief) the library user can see it when accessing components while browsing the library.

5 Summary

This report has described informational and technical documents for describing qualified components that are assets within a domain-specific reuse library. The glossy is a document intended to quickly provide a summation of various features of a component relative to a domain-specific library. The technical brief is an extension of the glossy providing basically the same information in greater detail. These documents are very important because they are basic elements in building a marketing strategy for supporting domain-specific library components. By following the standard formats outlined in this report, components are presented to library users in an easily recognizable format.

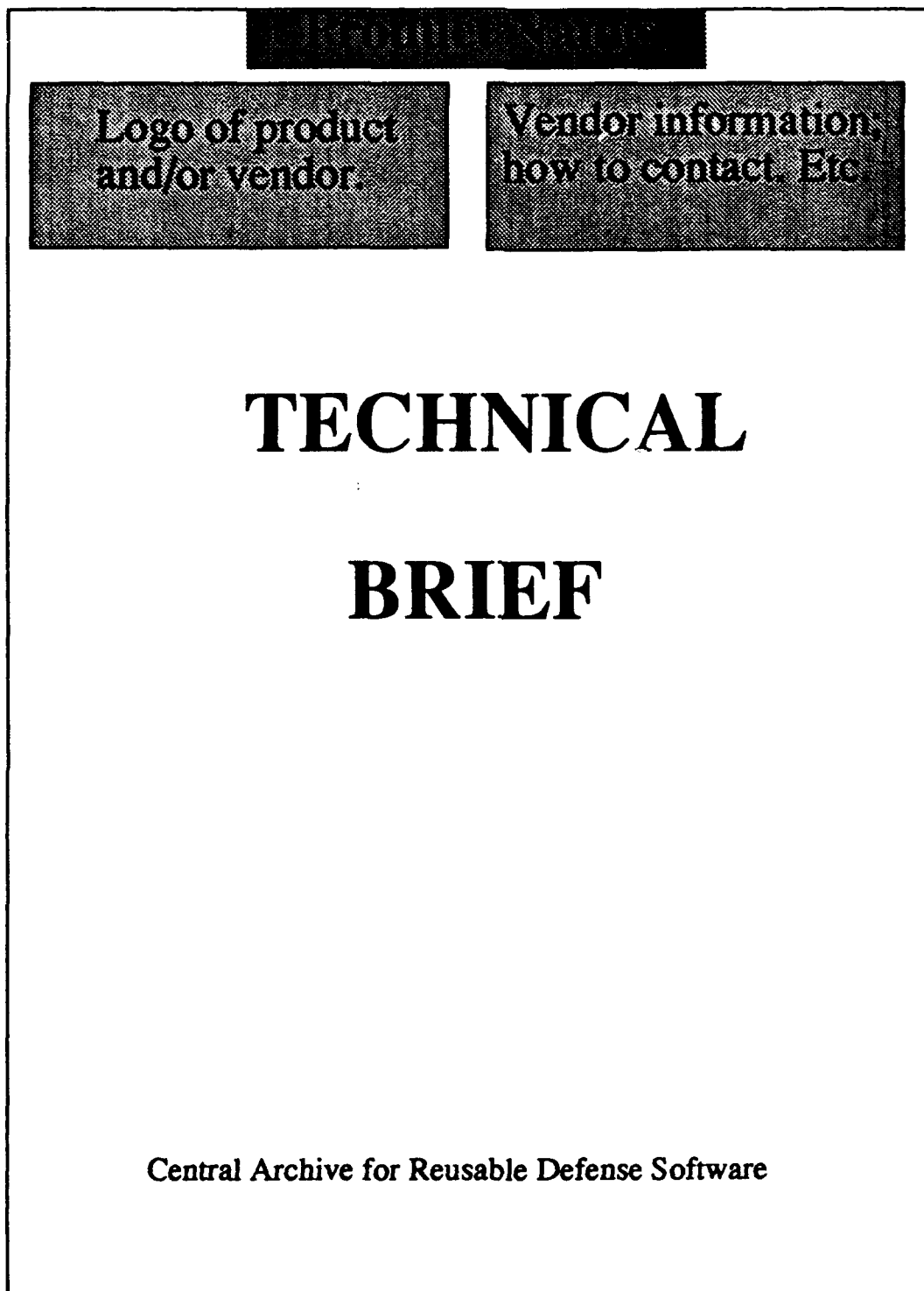
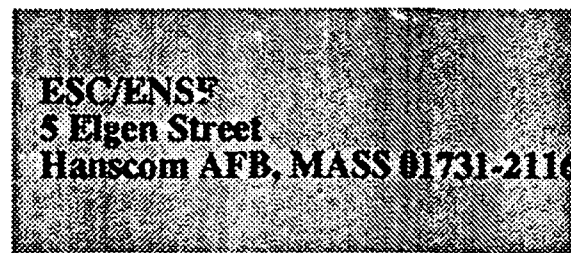
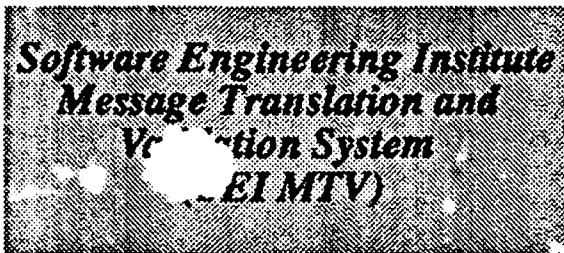


Figure 5-1. Cover for Technical Brief

APPENDIX A: Glossy examples

The following glossies are for demonstration purposes only. No claim is made as to the accuracy of the content. The focus is intended to be on the format and structure.



ROLE WITHIN THE DOMAIN

The Software Engineering Institute Message Translation and Validation (SEI MTV) system is a general model solution written in Ada, that can be used in a system when the system must convert between different representations of a message. These message representations can be character-based, bit-based, and internal.

MTV was written specifically for the Command, Control, Communications, and Intelligence domain where there is a need to translate and validate incoming and outgoing messages.

MTV consists of two parts, the first of which, termed the model solution, deals with templates and utilities supplied by the SF. These are used to produce packages called field and record typecasters and the definition of an external representation of the message called Interface Control Document (ICD). All the messages produced here are Ada reusable code. The second part of the MTV process takes these models and joins them together for a particular message type. This is an Ada package linked with the components produced above to execute the translation and validation of a message.

SALIENT FEATURES

Developers in other domains where a need to translate and validate messages exists, can understand and use the MTV model solution.

There is one version of MTV which handles either character or bit based messages, based on user input at the time the component is launched.

The MTV will perform its functions in real time, keeping up with the input message traffic. Based on typical command center requirements, MTV will handle at least 300 messages per minute on a 20 MIP workstation. The number of fields in the messages and whether they are character or bit based will influence the message throughput. Based on typical message complexity, MTV will handle the above message rate for either character or bit based messages for messages with at least 15 fields.

The SEI MTV has been used successfully by the USAF program Granite Sentry for character based messages.

TECHNICAL SPECIFICATIONS

Currently includes message typecasters for the bit_based message: Surveillance

The PRISM version of MTV includes message Typecasters for these ASCII messages: e3a; NUDET; Interceptor; 170 ROCC/SOCC The April MTV component was written partly in C and included Sybase specific SQL commands. It was then converted entirely to Ada, and the Sybase-specific SQL converted to ANSI SQL. This SQL translation required the development of the Database Broker component. MTV includes the typecasters and ICD generated from the SEI MTV templates and support software, and wrapper software written for PRISM. The wrapper software includes interface software to DECMessageQ (DMQ), SQL generation software, and the interface to the Database Broker.

The SEI MTV is written in Ada and contains no proprietary products. It was developed and tested on DEC/VMS with the intent that it could run on any platform. However, integration testing showed that some of its functionality had machine and compiler dependencies.

SUPPORT

Software produced by PRISM program at Electronic Systems Center(ESC), Systems and Software Design Center (AVS)

Capt. Paul Valdez, PRISM Deputy Program Manager: (617) 377-3458

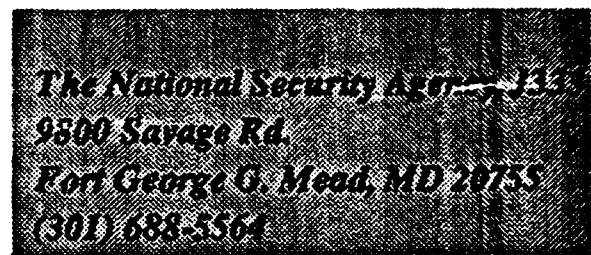
PRISM

ESC/AVSF

Hanscom AFB, MA 01731-2116

LICENSING AND AVAILABILITY

The SEI MTV product was written by the Software Engineering Institute (SEI). It is government off-the-shelf software and does not require a license.



ROLE WITHIN THE DOMAIN

Oilstock is designed to track airborne targets, including aircraft, satellites, and sea going vessels. In addition to data display, analysts can draw geositional overlays on OILSTOCK maps to produce reports or amplify information. OILSTOCK uses the CIA World Database II (WDBII) as its foundation for vector maps and the Defense Mapping Agency's ARC Digitized Raster Graphics for high quality raster maps. OILSTOCK reads Digital Terrain Elevation Data to conduct a visual line of site analysis.

Advanced features include ELINT/Direction Finding capabilities and calculating satellite footprints. OILSTOCK also supports cartographic overlays and numerous map projections. OILSTOCK reads data from a large number of data formats, including the CIA World Data Bank II (WDBII) and Digital Mapping Agency (DMA) digital maps.

SALIENT FEATURES

OILSTOCK is a stand-alone Geographic Information System (GIS) application with features that are present to better visualize and evaluate plots, including placement of icons, range rings, etc. onto a sketch pad which can be saved to a cartographic overlay, and draped over various map projections. OILSTOCK provides a mapping system application with support for analysis of target plotting. It is easily installed and evaluated with the help of well written manuals.

The various map projections supported by OILSTOCK allow numerous views of plotted data. The icons are already defined when delivered. Incorporation of new databases is described fully in the Oilstock Installation Guide and seems to be relatively uncomplicated.

TECHNICAL SPECIFICATIONS

OILSTOCK was developed in C on a System V UNIX machine, using X windows for display. It has been successfully compiled on the following platforms: Sun3, Sun4, Masscomp, Apollo, HP 720, Silicon Graphics, DEC, IBM RS/6000, and Data General workstations.

SUPPORT

Support is available from NSA via U.S. mail and a toll phone number:

The National Security Agency, J333

9800 Savage Rd.

Fort George G. Mead, MD 20755

(301) 688-5564

OILSTOCK is accompanied by excellent hard-copy manuals.

LICENSING AND AVAILABILITY

OILSTOCK has been released to the CARDS Program for Official Use Only (FOUO) by the National Security Agency (NSA). OILSTOCK has also been released to a number of other government agencies and contractors under the same restrictions.

APPENDIX B: Technical Briefs

The following technical briefs are for demonstration purposes only. No claim is made as to the accuracy of the content. The focus is intended to be on the format and structure.

SECRET

*Software Engineering Institute
Message Translation and
Validation System
(SEI MTV)*

ESC/ENSF
5 Elgen Street
Hanscom AFB, MASS 01731-2116

TECHNICAL BRIEF

Central Archive for Reusable Defense Software

INTRODUCTION/OVERVIEW

The Message Translation and Validation process, MTV, consists of two parts, the first of which, termed the *model solution*, is comprised of templates and utilities supplied by the Software Engineering Institute (SEI). These are used to produce packages called field and record typecasters, and the definition of an external representation of the message called the Interface Control Document, ICD, which is the second part of the MTV.

Application Area

The "item" described above is called the *validatory package*. All of the modules produced here are Ada reusable code. This part of the MTV process takes these modules and joins them together for a particular message type. This is an Ada package linked with the components produced above to execute the translation and validation of a message. It contains the executive `gcc_validate` and its supporting functions and procedures. It is called by the System manager, `gcc_sysmgr`, to execute the translation and validation of an incoming message. All the record typecasters and ICDs for the message types supported are linked into the `gcc_validate` procedure. This module brings together the two SEI solutions produced above for a particular message type, the record typecasters and the ICD. The translation and validation of a message consists of a transformation from an input ASCII string with field separators to a byte array internal image. Then this array becomes an Ada enumeration where, for example, the input character 'H' of a track class field has been transformed to its expanded equivalent 'HOSTILE'. Finally, the enumeration image is transformed into the result user representation, an ASCII character string describing the expanded input message. These image transformations are accomplished through three function calls: one in the ICD to extract the first image; two in the record typecasters to achieve the result string thus the typecasters, the ICD; and the executive are joined. The result string is then parsed onto a SQL command line for insert. The client module, `gcc_sysmgr` is then called to access the SQL server to store translated messages in the actual or exercise database. The MTV component performs the translation and validation of both characters and bit based messages and the update of the mission database the mission application displays.

Limitations

The executive is coded manually but a template can be made in the future to facilitate its coding

DETAILED FUNCTIONAL DESCRIPTION

SEI MTV is a model that is a general solution, written in Ada, that can be used in a system where the system must convert between different representations of a message. These message representations can be character-based, bit-based, and internal. MTV was written specifically for the Command, Control, Communications, and Intelligence domain where there is a need to translate and validate incoming and outgoing messages.

The MTV consists of two parts, the first of which, termed the *model solution* deals with templates and utilities supplied by the SEI. These are used to produce packages called field and record typecasters and the definition of an external representation of the message called ICD. All of the messages produced here are Ada reusable code. The second part of the MTV process takes these models and joins them together for particular message type. This is an Ada package linked with the components produced above to execute the translation and validation of a message.

SPECIFIC FUNCTIONS WITHIN THE DOMAIN

The Generic Command Center Architecture (GCCA), as defined in the PRISM GCCA report, has a requirement for a MTV component which does message translation and validation of character and bit-based messages, with subsequent updates to the database and mission application. These are primarily fixed format messages, although variant fields and fixed repetitions of fields are included, e.g., ROCC/SOCC and Tactical Digital Information Link (TADIL) messages. Messages with highly complex and variable structures, e.g., United States Message Text Format(USMTF) messages, are not included under MTV, but are handled by the Automated Message Handler(AMH) component. It is also part of inter-process communications within a local area network.

DESCRIPTION OF SALIENT FEATURES

Developers in other domains (e.g., intelligence) where a need to translate and validate messages exist, can understand and use the MTV model solution. There is one version of MTV which handles either character or bit based messages, based on user input at the time the component is launched. The MTV will perform its functions in real time, keeping up with the input message traffic. The number of fields in the messages and whether they are character or bit based will influence the message throughput. Based on typical message complexity, MTV will handle the above message rate for either character or bit based messages with at least 15 fields. The SEI MTV has been used successfully by Granite Sentry for character based messages.

The SEI MTV uses a model solution that can be used in a system to convert between different representations of a message. These message representations can be character based, bit based, internal(Ada values), or the ASCII equivalent of the message. The MTV software consists of a set of utilities and Ada coding templates. An MTC coding template is an Ada file containing incomplete, ie, non-compilable code. Each template includes a package specification, package body, and a test procedure which tests the functionality implemented by the completed template. The templates contain placeholders, by which the message fields and the ICD are defined uniquely. The definition of a message field requires a typecasters template. The field type, e.g., integer, enumeration, determines the particular typecasters template to use. The definition of the entire message requires a record typecasters. The completed typecasters requires a record typecasters. The completed typecasters templates are compilable Ada packages, called typecasters. The external view of the message is built from an ICD template, where the term ICD alludes to Interface Central Document. The message type, character or bit, determines the particular ICD template to use. The completion of this template results in a field-by-field description of the message in terms of size, type, separator, bit position, etc., depending on whether the message is character or bit based. The completed templates are compilable Ada packages, called ICDs. An ICD depends on the typecasters packages corresponding to the message fields and on the record typecasters package that defines the message.

TECHNICAL SPECIFICATIONS

SEI MTV currently includes message typecasters for the bit_based message: Surveillance

The PRISM version of MTV includes message typecasters for these ASCII messages: e3a; NUDET, Interceptor; 170 ROCC/SOCC. The April MTV component was written partly in C and included Sybase specific SWL commands. It was then converted entirely to ADA and the Sybase-specific SQL converted to ANSI SQL. This SQL translation required the development of the Database Broker component. MTV includes the typecasters and ICD generated from the SEI MTV templates and support software, and

wrapper software written for PRISM. The wrapper software includes interface software to DEC MessageQ(DMQ), SQL generation software, and the interface to the Database Broker. The SEI MTV is written in Ada and contains no proprietary products. It was developed and tested on DEC/VMS with the intent that it could run on any platform. However, integration testing showed that some of its functionality had machine and compiler dependencies.

With the addition of some wrapper software and modifications to the SEI MTV fulfilled the component requirements. These modifications were required primarily to handle machine and compiler dependencies. The wrapper software interfaced the product to other GCCA components on the mission application. The UNIX tools were found to be general purpose enough to handle very complex character based messages, but could not handle bit_based messages. The use of these tools would require significant software development.

SUPPORT

Software produce by PRISM program at Electronic Systems Center(ESC), Systems and Software Design Center (AVS)

Capt. Paul Valdez, PRISM Deputy Program Manager: 617-377-3458

PRISM

ESC/AVSI

Hanscom AFB, MA 01731-5000

LICENSING AND AVAILABILITY

The SEI MTV product was written by the Software Engineering Institute(SEI). It is government off-the-shelf software and does not require a license.

OLDS TOUGHS

*National
Security Agency*

*The National Security Agency, 1333
9800 Savage Rd.
Fort George G. Mead, MD 20755
(301) 688-5564*

TECHNICAL BRIEF

Central Archive for Reusable Defense Software

INTRODUCTION/OVERVIEW

OILSTOCK is a high resolution interactive graphics system. In addition to data display, analysts can use it to draw geositional overlays on OILSTOCK maps to produce reports or amplify information.

OILSTOCK uses the CIA World Database II (WDBII) as its foundation for vector maps and the Defense Mapping Agency's ARC Digitized Raster Graphics for high quality raster maps. OILSTOCK reads Digital Terrain Elevation Data to conduct a visual line of site analysis.

Application Area

OILSTOCK is accompanied by excellent hard-copy manuals, which make installation quite easy. The organization makes them easy to use as a reference and a tutorial. An on-line help facility is also available. It is recognized that the specific functionality requirements of command centers for their spatial information capabilities vary greatly. Further, given the realization of the diverse nature of the different approaches taken for packaging GIS and mapping system components, and its subsequent evaluation, it is necessary to be cognizant of these differences when defining the evaluation criteria. Therefore, the tact taken in evaluating such systems will focus on a very small set of core or critical criteria. Other features are examined and noted, but generally deemed non-critical for inclusion into the CARDS Command Center Library. It is then left to the library user to examine the report to determine if a particular system meets the user's requirements.

Limitations

The interface is somewhat cumbersome to use at first, although it becomes substantially easier as it is used. In some cases during the qualification process of OILSTOCK into the CARDS Command Center Library, odd projections and views in conjunction with user defined overlays produced some unexpected results. It would seem some defects are present, but it was easy to undo the defective map and try again in another manner. No other anomalies were discovered..

DETAILED FUNCTIONAL DESCRIPTION

OILSTOCK uses the CIA World Database II (WDBII) as its foundation for vector maps and the Defense Mapping Agency's ARC Digitized Raster Graphics for high quality raster maps. OILSTOCK reads Digital Terrain Elevation Data to conduct a visual line of site analysis. OILSTOCK also supports cartographic overlays and numerous map projections. According to the manual, each of these projections exists to support different human-based analysis. The plots and overlays are adjusted to reflect the projection displayed. For example, if an overlay is created with an icon of the Empire State Building placed on New York City, then scrolling, zooming and changing the map projection does not affect the placement of the icon, i.e. it is still on New York City. However, the size of the icon remains constant. Icons may be placed by latitude/longitude specification or by point - and- click mouse interaction. Icons may not be used in a target plot. Target position is updated via messages sent to the OILSTOCK system. ASCII files in a specified format can also be used to display tracking information. OILSTOCK is also used to display tracking information. OILSTOCK does not allow either on demand or autonomous DBMS access.

SPECIFIC FUNCTIONS WITHIN THE DOMAIN

Oilstock is designed to track airborne targets, including aircraft, satellites, and sea going vessels. In addition to data display, analysts can Draw geositional overlays on OILSTOCK maps to produce reports or amplify information.

DESCRIPTION OF SALIENT FEATURES

OILSTOCK is a stand-alone Geographic Information System (GIS) application with features that are present to better visualize and evaluate plots, including placement of icons, range rings, etc. onto a sketch pad which can be saved to a cartographic overlay, and draped over various map projections. OILSTOCK provides a mapping system application with support for analysis of target plotting. It is easily installed and evaluated with the help of well written manuals. The various map projections supported by OILSTOCK allow numerous views of plotted data. The icons are already defined when delivered. Incorporation of new databases is described fully in the Oilstock Installation Guide and seems to be relatively uncomplicated.

Advanced features include ELINT/Direction Finding capabilities and calculating satellite footprints. OILSTOCK also supports cartographic overlays and numerous map projections. OILSTOCK reads data from a large number of data formats, including the CIA World Data Bank II (WDBII) and Digital Mapping Agency (DMA) digital maps. The many and varied GIS and mapping system commercial and government products take two basic approaches to how they are packaged. First is the stand-alone application, where a user's interaction is limited to a graphical user interface (GUI). Other systems are packaged as a toolkit, where the functionality of the system is available to the user via application programming interfaces (API) special scripting languages, and macro construction, which can be used in concert to create an application. Most often it is a hybrid, usually leaning more toward the stand-alone system, a system with some flexibility in interaction, but functioning primarily as a single stand-alone application. OILSTOCK is not a GIS tool-kit, but is rather a completed GIS application that might be constructed from such a tool-kit.

TECHNICAL SPECIFICATIONS

OILSTOCK was developed in C on a System V UNIX machine, using X windows for display. It was also shown to work properly under Motif and TWM. These latter two environments were not as thoroughly tested. It has been successfully compiled on the following platforms: Sun3, Sun4, Masscomp, Apollo, HP 720, Silicon Graphics, DEC, IBM RS/6000, and Data General workstations.

SUPPORT

Support is available from NSA via U.S. mail and a toll phone number:

The National Security Agency, J333

9800 Savage Rd.

Fort George G. Mead, MD 20755

(301) 688-5564

OILSTOCK is accompanied by excellent hard-copy manuals.

LICENSING AND AVAILABILITY

OILSTOCK has been released to the CARDS Program for Official Use Only (FOUO) by the National Security Agency (NSA). OILSTOCK has also been released to a number of other government agencies and contractors under the same restrictions.

REFERENCES

- [1] DoD Domain Definition Report, DoD Center for Software Reuse Operations, The Defense Information Systems Agency/Corporate Information Management (DISA/CIM), May 92.
- [2] Huber, Theresa R., "Findings of the CARDS Sponsored Software Reuse Legal Workshop", DSD Laboratories, Inc., Proceedings of the 6th Annual Workshop on Software Reuse (WISR93), Owego, NY, 2-4 Nov 93
- [3] Library Development Handbook. Informal Technical Report STARS-VC-B005, 29 October 1993.
- [4] Library Operations Policies and Procedures for the Central Archive for Reusable Defense Software: Update - Volumes I, II, III. Informal Technical Report STARS-VC-B004/002/00, B004/002/01; Vol. I and Vol. II, 27 February 1994
- [5] STARS Reuse concepts, Volume I, Conceptual Framework for Reuse Processes (CFRP), Version 2, STARS-UC-05159/00100, 13 Nov 92.
- [6] Poore, J.H. and Theresa Pepin. "Criteria and Implementation Procedures for Evaluation of Reusable Software Engineering Assets," ASSET The National Software Technology Repository, March 1992.
- [7] Technical Concept Document, Central Archive for Reusable Defense Software (CARDS), STARS-VC-03536/001/00, 27 February 1994.
- [8] Tracz, Will. "A conceptual model for megaprogramming," Software Engineering Notes, ACM Press, Vol. 16, Number 3 (July 1991).
- [9] Wallnau, Kurt C. CARDS Libraries and a Standard DoD Component Classification Scheme. Paramax Systems Corporation, February 1993.